Computational Security

Cryptography and Protocols
Andrei Bulatov

Perfect Security

- Let (K,E,D) be a symmetric encryption scheme. It is said to be perfectly secure if for any two plaintexts P_1, P_2 and a ciphertext C

$$\Pr[E_k(P_1) = C] = \Pr[E_k(P_2) = C],$$

where the probability is over the random choice k ← K, and also over the coins flipped by E

Perfect Security is Far From Perfect

不切实际的

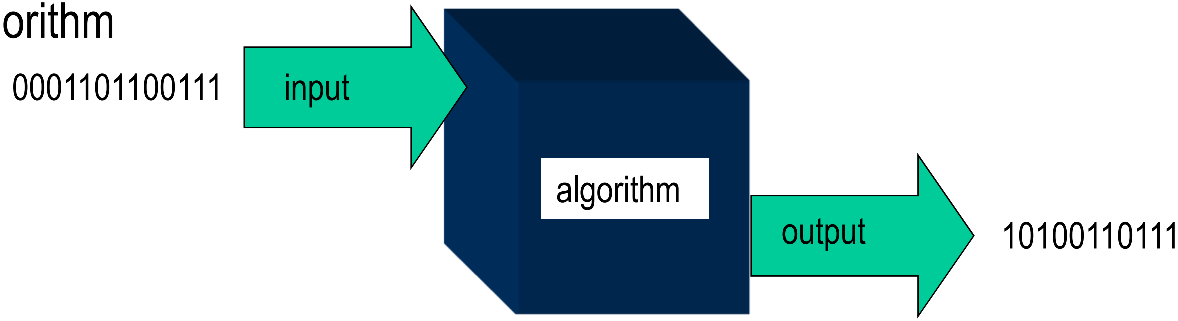
- Perfect security is impractically strong
- Need to relax it. There are two ways to do that

- *Put restrictions on the adversary*: not almighty, but computationally efficient

- *Give the adversary some chance*: do not insist that the probabilities in the definition are equal, just close

Algorithms

- Algorithm



- Algorithm performs a sequence of `elementary steps'   that can be:
  - arithmetic operations
  - bit operations
  - Turing machine moves
  - ……….   (but not quantum computing!!)
- We allow probabilistic algorithms, that is, flipping coins is permitted

use some random bit for its computation

Complexity

- The time complexity of algorithm A is function $f(n)$ that is equal to the number of elementary steps required to process the most difficult input of length n worst case complexity
- We do not distinguish algorithms of complexity $2n^2$ and $100000n^2$ 忽略系数
- A computational problem has time complexity at most $f(n)$ if there is an algorithm that solves the problem and has complexity $O(f(n))$ f(n) 是上限 ceiling

  - problem solvable in linear time: there is an algorithm that on input of length n performs at most Cn steps

  - problem solvable in quadratic time: there is an algorithm that on input of length n performs at most Cn^2 steps

Complexity (cntd)

- Polynomial time solvable problems: $\Longleftrightarrow$ Efficient

  There is a polynomial $p(n)$ such that the problem is solvable in time $O(p(n))$

- P - class of problems solvable in poly time by a deterministic algorithm

- BPP - class of problems solvable in poly time by a probabilistic algorithm Bounded Probabilistic Polynomial

- An algorithm is superpolynomial if its time complexity $f(n)$ is not in $O(p(n))$ for any polynomial $p(n)$

  not efficient

  k letters → k! attempt → not poly → super-poly

  eg. try all possible permutations of the alphabet (Brute Force)

- A function $\varepsilon: N \rightarrow [0,1]$ is polynomially bounded if $\varepsilon(n) \geq \frac{1}{p(n)}$ for some polynomial $p(n)$

  ε is a very small number

  eg. $\varepsilon(n) \geq \frac{1}{n^3}$

Usually Brute Force is inefficient

Statistical Tests

窃听者

- In the definition of security as a game the Eavesdropper sees a ciphertext C that is an encryption of one of the two plaintexts
- The only thing Eve can do is to run some algorithm on C that tells her whether it is an encryption of P_1 or P_2
- Such algorithms are called statistical tests
- More formally, a statistical test is an algorithm (function) that on input from $\{0,1\}^n$ outputs 0 or 1

  0 – the algorithm thinks the ciphertext is P_1
  1 – the . . . . is P_2
- Examples:
  - ❖ On input $C \in \{0,1\}^n$ output 1 if the second byte of C is 00 (Hex), otherwise output 0 or 1 with probability 1/2
  - ❖ On input $C \in \{0,1\}^n$ output 1 if C contains a string of consecutive 0s or 1s of length at least $\log(n) + 1$, otherwise 0

    ↳ distinguishes "human generated bit string" & "real random bit strings"

Indistinguishability 不可区分性

prob.

- Let W_1, W_2 be two distributions on $\{0,1\}^n$.
- Distributions W_1, W_2 are said to be computationally indistinguishable if for any efficient statistical test Eve

$$\left| \Pr_{X \leftarrow W_1}[Eve(X) = 1] - \Pr_{X \leftarrow W_2}[Eve(X) = 1] \right| < \varepsilon$$

where $\Pr_{X \leftarrow W_i}$ means that X is sampled from W_i, and ε is negligible. 微不足道的

- ε is often called the advantage of the test

$\varepsilon = 0 \rightarrow$ abs. random
$\varepsilon = 1 \rightarrow$ abs. correct
注：这俩不是一个概念

- Need to clarify several things:
  - What is an efficient test
  - What is negligible ε
  - How is it related to security

语义安全

Computational (Semantic) Security

- Distributions that appear naturally in cryptography are encryptions of some plaintext with a random key.
- Let P be a plaintext and the corresponding distribution W over all possible ciphertexts that are produced from P assigns a ciphertex C the probability

$$\Pr_{k \leftarrow K}[E_k(P) = C]$$

- An SES (K, E, D) is said to be computationally secure if for any plaintexts P_1, P_2 the corresponding distributions W_1, W_2 are computationally indistinguishable

Computational Security as a Game

- We assume that Eve is efficient
- Game
  - Alice chooses a key k
  - Eve chooses 2 plaintexts and gives them to Alice
  - Alice encrypts one of them and sends to Eve
  - Eve decides which one is encrypted

  Eve wins if her decision is right
- The system is computationally secure if Eve wins with probability $\frac{1}{2} + \varepsilon$, where ε is negligible

Efficient Statistical Tests

- In practice: Takes reasonable time to run
- Difficulties: What time is reasonable? Different grades of security

  We do not know the adversary's capabilities
- In theory: Adversary is polynomial time 256^3 还是 efficient 的
- This means that we consider the adversary's performance dynamically, looking how it scales as the length of keys and messages grow 在 big data 里, 平方都慢

Negligible Advantage

- What ε is negligible?
- In practice:
- Depends on the task, but generally $\varepsilon = 2^{-30}$ is not negligible. It is about one billionth and is comparable with the amount of data we have to deal with
- $\varepsilon = 2^{-100}$ is negligible. At least for now. With the current technology there is no way the adversary has anything close to 2^{100} $\approx 10^{27}$ attempts on your cryptosystem. But things change …
- In theory: The advantage is not polynomially bounded, that is, ε decreases faster than $\frac{1}{p(n)}$ for any polynomial p